In the Drawings:

FIGs. 7, 11A and 11B, were objected to due to improper margin spacing. Applicants have provided two drawing replacement sheets to correct the margins of FIGs. 7, 11A and 11B.

REMARKS

Claims 44-56 have been amended to change the term "carrier medium" to "computer readable medium."

Reconsideration is respectfully requested in light of the following remarks.

Specification Objection:

The Abstract was objected to for exceeding 150 words. The Abstract has been amended to contain no more than 150 words.

Drawing Objection:

Figures 7, 11A and 11B were objected to because of improper margin sizes. Two replacement sheets are included herewith to overcome this objection.

Section 103(a) Rejection:

The Office Action rejected claims 1-12, 15-27, 30-40, 43-53 and 56 under 35 U.S.C. § 103(a) as being unpatentable over Aman et al. (U.S. Patent 6,694,346) (hereinafter "Aman") in view of Knudsen (U.S. Patent 5,682,535). Applicants respectfully traverse this rejection in light of the following remarks.

Regarding claim 1, contrary to the Examiner's assertion, Aman in view of Knudsen fails to teach wherein the in-memory heap *comprises a cached portion* of the store heap for the process. Aman teaches a system that enables a long running, reusable, virtual machine including a shared heap for stores runtime code for initializing a virtual machine and a private heap for application use (Aman, Abstract). The Examiner cites Figures 3 and 4 of Aman, but Applicants note that neither Figure 3 nor Figure 4 includes anything regarding a *cached* portion of the store heap. Figure 3 illustrates a multi-heap,

multi-process virtual machine including a shared heap and two private heaps and Figure 4 illustrates a private heap in detail. Neither the cited figures, nor the accompanied description mention anything about an in-memory heap comprising a cached portion of a store heap (Aman, Figures 3 and 4, column 4, lines 22-58).

Additionally, Aman in view of Knudsen also fails to teach performing an atomic transaction on the virtual heap, wherein said performing the atomic transaction comprises performing one or more transaction tasks, and wherein said performing the atomic transaction changes a state of the virtual heap by modifying one or more portions of the virtual heap. The Examiner's cited passage (Aman, column 4, lines 10-13) describes the read-only and read/write portions of Aman's shared heap, but fails to mention anything about performing atomic transactions on the virtual heap. The cited passage discloses using lock protections to prevent conflicts when two different processes are both modifying the shared heap (Aman, column 4, lines 10-13), but Applicants note that using inter-process locking mechanisms is very different from performing atomic transactions on the virtual heap. In fact, nowhere does Aman disclose performing atomic transactions on the virtual heap. Knudsen only teaches atomic database transactions for database table updates using specific COMMIT and ROLLBACK statements executable by his specific virtual stack machine (Knudsen column 14, lines 40-48). But, Knudsen fails to teach anything about a virtual heap and thus fails to disclose or suggest performing atomic transactions on a virtual heap.

Further regarding claim 1, Applicants submit that, contrary to the Examiner's assertion, Knudsen fails to disclose committing the atomic transaction by accepting the modifications to the one or more portions of the *virtual heap* if the one or more transaction tasks in the atomic transaction are performed without generating an error; and rejecting the atomic transaction by restoring the *virtual heap* to the state of the virtual heap prior to the performing the atomic transaction if one or more of the one or more transaction tasks in the atomic transaction generates an error when preformed. In fact, Knudsen fails to teach anything regarding a virtual heap. In contrast, Knudsen teaches a virtual stack machine that executes on a specific HURON data processing system and

that executes object code level representations of rules in order to read and write database data tables (Knudsen, column 4, lines 16-58, and lines 44-48). Knudsen teaches COMMIT and ROLLBACK statements used in his rule processing virtual stack machine to provide transaction synchronization points for changes applied to database tables (Knudsen, column 14, lines 40-45). However, Knudsen fails to teach committing or rejecting atomic transactions on a virtual heap for a process *executing within a virtual machine*. Thus, both Aman and Knudsen, separately and in combination, fail to teach committing and rejecting atomic transactions for managing a virtual heap for a process executing within a virtual machine.

Furthermore, Knudsen's system employs a data access method that is optimized for a unique access structure, consisting of tables, rows and fields, of his database and that performs all the access functions on the database (Knudsen, column 2, lines 1-26). Such a specific access method has nothing in common with Aman's method of virtual memory management for processes executing in reusable virtual machines. Additionally, Knudsen's operating system "operates based on an isomorphic programming representation that requires a translator/detranslator in order to perform programming development (Knudsen, column 2, lines 49-61). Knudsen's system simply does not apply to Aman's virtual machines since Knudsen utilizes a specific virtual stack machine only capable of executing very specific object code representations of database access rules on very a specific data processing system. Such a system is clearly incompatible with Aman's system that "includes a set of functions that behave in a consistent manner regardless of the hardware or operating system on which the virtual machine is running" and where applications "need not be aware of any operating system or platform inconsistencies or unique functionalities" (Aman, column 3, lines 26-32). Thus, the teachings of Knudsen do not suggest any modifications to the system of Aman.

Applicants further submit that the proposed combination of Aman and Knudsen, even if suggested, would not result in a system that would include the limitations of Applicants' claim 1. Specifically, such a combination would result in a rule-based, virtual stack machine that includes the ability to create reusable virtual stack machines.

Such a combination of Aman and Knudsen would fail to disclose include providing an in-memory heap <u>comprising a cached portion</u> of a store heap; <u>performing an atomic transaction on the virtual heap</u>; committing the atomic transaction by <u>accepting the modification to the one or more portions of the virtual heap</u>; and rejecting the atomic transaction by <u>restoring the virtual heap</u> to the state of the virtual heap prior to said performing the atomic transaction. Neither reference, alone or in combination, suggests these features.

For at least the reasons given above, the rejection of claim 1 is not supported by the prior art at removal of the rejection is respectfully requested. Similar remarks as discussed above in regard to claim 1 apply to claim 44.

Regarding claim 16, contrary to the Examiner's assertion, Aman in view of Knudsen fails to disclose wherein the in-memory heap <u>*comprises a cached portion*</u> of the store heap for the first process. As discussed above regarding the rejection of claim 1, Aman teaches a system that enables a long running, reusable, virtual machine including a shared heap for stores runtime code for initializing a virtual machine and a private heap for application use (Aman, Abstract). The Examiner cites Figures 3 and 4 of Aman in support of his assertion regarding Aman teaching a cache heap, but Applicants note that neither Figures 3 nor Figures 4 includes anything regarding a cached portion of the store heap. Figures 3 illustrates a multi-heap, multi-process virtual machine including a shared heap and two private heaps and Figures 4 illustrates a private heap in detail. Neither the cited figures, nor the accompanied description mention anything about an in-memory heap comprising a cached portion of a store heap. Additionally, as discussed above, Knudsen fails to teach anything regarding virtual heaps and clearly fails to mention anything about a cached portion of an in-memory heap.

Further regarding claim 16, contrary to the Examiner's assertion, Aman in view of Knudsen fails to teach providing an application programming interface (API) <u>for performing heap operations on the virtual heap</u>, wherein the API comprises functions for performing operations on portions of the virtual heap, and wherein the functions in the

API are callable by processes *executing within the virtual machine.* The Examiner admits that Aman fails to disclose such an API, but argues that Knudsen teaches providing an application programming interface. However, Knudsen does not teach providing an API for performing heap operations on a virtual heap. The Examiner's cited passage refers to a table access method that provides an interface between the virtual stack machine and the data storage system (Knudsen, column 4, lines 2-4). Knudsen teaches that his system "provides an operating system, data base, and access method which pushes data definitions, input editing and validation, selection and ordering, looping, output editing and validation, error processing, and auditing down *into the data access method*" (Knudsen, column 1, lines 61-67). Knudsen further teaches that his system includes an "access structure [that] consists of a plurality of tables, each table having a plurality of rows, and each row having a plurality of fields" and that "each row in the access structure is identified by a unique primary key in one of the fields of the row and by a table identifier" (Knudsen, column 2, lines 16-26). Knudsen also describes how his "access method maintains indexes into the tables stored in the access structure" (Knudsen, column 2, lines 26-30). Thus, Knudsen teaches a data access method that maintains very specific data structures for access database tables that are incompatible with, and have nothing in common with, performing operations on a virtual heap.

Knudsen does not provide an API comprising functions for performing operations on portions of a virtual heap, and certainly does not teach providing such an API where the functions of the API are callable by processes executed within the virtual machine. As noted above, Knudsen fails to teach anything regarding memory management for processes executing on a virtual machine. Thus, Aman in view of Knudsen fails to teach providing an API for performing heap operations on the virtual heap, wherein the API comprises functions for performing operations on portions of the virtual heap, and wherein the functions in the API are callable by processes executing within the virtual machine.

Further regarding claim 16, Applicants submit that, contrary to the Examiner's assertion, Knudsen fails to disclose committing the first operation on the first portion of

the virtual heap by accepting the modification to the first portion of the virtual heap if the first operation is performed without generating an error; and rejecting the first operation on the first portion of the virtual heap by restoring the virtual heap to the state of the virtual heap prior to said performing the first operation if the first operation generates an error when performed. Knudsen teaches COMMIT and ROLLBACK statements used in his rule processing virtual stack machine to provide transaction synchronization points for changes applied to database tables (Knudsen, column 14, lines 40-45). However, Knudsen fails to teach committing or rejecting atomic transactions on a virtual heap for a process *executing within a virtual machine.* As discussed above regarding the rejection of claim 1, Knudsen's system is wholly incompatible with Aman's. Thus, both Aman and Knudsen, separately and in combination, fail to teach committing and rejecting atomic transactions for managing a virtual heap for a process executing within a virtual machine.

For at least the reasons given above, the rejection of claim 16 is not supported by the prior art at removal of the rejection is respectfully requested.

Regarding claim 31, contrary to the Examiner's assertion, Aman in view of Knudsen fails to teach wherein the in-memory heap comprises cached portions of the store heap for access by the process. As shown above regarding the rejections of claims 1 and 16, Aman fails to disclose an in-memory heap comprising cached portions of a store heap. Knudsen also fails to teach cached portions of a store heap because Knudsen, as shown above, fails to teach anything at all regarding a virtual heap or heaps in general.

Additionally, contrary to the Examiner's assertion, Aman in view of Knudsen fails to teach wherein a device is configured to perform operations on the virtual heap according to an application programming interface (API), and wherein the API comprises functions for performing the operations on the virtual heap. The Examiner admits that Aman fails to teach providing an API for performing operation on the virtual heap, and as described above regarding the rejections of claim 1 and 16, Knudsen also fails to teach providing such an API. Thus, both Aman and Knudsen fail to disclose a device

configured to perform operations on the virtual heap according to an application programming interface (API) comprising functions for performing the operations on the virtual heap.

Further regarding claim 31, and similar to the discussion above regarding claims 1 and 16, Aman in view of Knudsen fails to teach an API configured to: perform an atomic transaction on the virtual heap, wherein the atomic transaction comprises one or more transaction tasks, and wherein the atomic transaction changes a state of the virtual heap by modifying one or more portions of the virtual heap. Aman fails to teach anything regarding atomic transactions and Knudsen only COMMIT and ROLLBACK statements used in his rule processing virtual stack machine to provide transaction synchronization points for changes applied to database tables (Knudsen, column 14, lines 40-45). However, Knudsen fails to teach an API configured to perform an atomic transaction on the virtual heap. Knudsen's COMMIT and ROLLBACK action statement are executed by rules that are, in effect, "extended case statement[s]" (Knudsen, column 8, lines 22-24). Hence, Knudsen is not teaching an API configured to perform an atomic transaction on a virtual heap, but rather is including COMMIT and ROLLBACK action statement that may be specified as in rule-based case statements and executed by his virtual stack machine by a HURON data processing system.

Additionally, contrary to the Examiner's assertion, Aman in view of Knudsen fails to teach wherein the API is configured to commit the atomic transaction by accepting the modification to the one or more portions of the virtual heap if the one or more transaction tasks in the atomic transaction are performed without generating an error. Aman in view of Knudsen also fails to teach wherein the API is configured to reject the atomic transaction by restoring the virtual heap to the state of the virtual heap prior to the atomic transaction if one or more of the one or more transaction tasks in the atomic transaction generates an error when performed. Aman, as admitted by the Examiner, fails to teach atomic transactions and, as described above, Knudsen's COMMIT and ROLLBACK statement have nothing to do with a virtual heap or with accepting modifications to one or more portions of a virtual heap.

For at least the reasons given above, the rejection of claim 31 is not supported by the prior art at removal of the rejection is respectfully requested.

Claims 13-14, 28-29, 41-42 and 54-55 were rejected 35 U.S.C. § 103(a) under as being unpatentable over Aman in view of Knudsen and further in view of Lawrence (U.S. Patent 6,629,113). The rejection of these claims is unsupported by the cited art for at least the reasons given above in regard to their respective independent claims.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion in regard to the dependent claims is not necessary at this time.

Information Disclosure Statement:

Applicants note that information disclosure statements with accompanying Forms PTO-1449 were submitted on June 24, 2004 (electronically) and August 19, 2004. Applicants request the Examiner to carefully consider the listed references and return copies of the signed and initialed Forms PTO-1449 from each statement. Applicants are enclosing copies of the PTO-1449 forms for the Examiner's convenience.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicants hereby petition for such extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-46700/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

☐ Petition for Extension of Time

☒ Copies of previously submitted forms PTO-1449

☒ Drawing Replacement Sheets

☐ Fee Authorization Form authorizing a deposit account debit in the amount of $ for fees ( ).

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: October 14, 2004